下载APP

23 | 考虑限制，让自己的产品不入险地

2020-12-16 乔新亮

乔新亮的CTO成长复盘 进入课程 >

讲述：乔新亮
时长 24:29 大小 22.43M

你好，我是乔新亮。

在实际的工作中，可能你经常会承担许多难以按约、按时交付，甚至是无法交付的工作。对于个人成长来说，这无疑是个大问题。

虽然我常常向团队强调一个公式：「**认知到位 + 彪悍执行 = 成功交付**」，但这是建立在对项目的客观评估基础上的，否则，对于某些工作任务或产品需求来说，无论你多么努力，也可能无法保质保量地交付。

无法交付的原因可能有很多：或许，该需求远远超出了团队现阶段的技术能力；又或许，是该需求违背了项目执行的客观规律，等等。总之，因为你承诺了自己根本做不到，或风

险极高的“契约”，因而导致自己无法按约交付，最终对自己的成长和信誉造成了损害。当然了，对于公司而言，这一结果也是最差的。

举一个夸张点的例子，有一天，老板找到你说，小王啊，公司最近要自研一款分布式数据库，决定让你来负责项目实施，一个月能不能搞定上线？

你可能会在心里想，天啊，一个月，时间太紧了，怎么可能？老板看出你有点犹豫，又补充道：小王啊，要抓住机会，好好表现啊！

你一听，顿时觉得，不就是加班吗？拼一把吧，于是点头应承下来。结果，一个月后，项目因为种种意外延期了，你则因为长期加班累了个半死……

我们不妨再举一个例子，公司的新产品要上线了，老板和产品经理们一同规划了一艘“航空母舰”：功能极其强大、性能业界领先、UI 非常漂亮，同时希望一个月就上线。结果呢，团队奋战了几个月，不但产品的设计和研发处处受阻，上线时间还遥遥无期……

以上两个例子，相对来说都比较夸张，但夸大是为了便于理解。类似的情况，其实每天都在各大公司内上演，只是更加隐蔽，不易察觉。你可能认为这是一个有关“自知之明”的问题：没有金刚钻就别揽瓷器活，自己能干多少活还不清楚吗？

但我要说，**无论是什么问题，只要存在，就一定有其合理性；进一步讲，无论是什么问题，只要合理，就一定有专业的解决办法，因为那些专业人士总是能确保自己不掉入陷阱。**

在这里，我们同样有一套专业的思考和解决方案，适用于架构设计、个人成长等诸多方面，它的名字，叫作“考虑限制”。

其实，在前面的几讲内容里，我们已经多少聊过一点“限制”思想。比如，对于高可用架构设计来说，要做好流量控制，否则，所谓的“高可用”只能是空谈；对于产品设计来说，要明确给用户的契约，不能做无边界的承诺。

但关于“限制”，我们还需要更加系统化的理解和实践。接下来，我们就来详细聊聊，如何做好限制，让产品不入险地。

限制产品的输入与输出

首先，我们要知道，只有专业的人，“懂”的人才能做好限制，这个道理在高可用、高并发、高性能等各个领域都是通用的。

那么，如何才能变得专业，让自己跨越从“不懂”到“懂”的鸿沟呢？这里就再次应用到了架构设计思维，对应的第一步，就是做好拆解。

任何产品都存在“输入”和“输出”两部分。如果你是产品的负责人，“输入”就是其他人对你承诺的“契约”；“输出”一般理解为承诺给用户的“契约”，指的是产品的对外能力，通常受到输入的严重影响。

比如，因为资金有限、服务器不够，所以我们无法承诺服务器在 300 万并发压力下保证 TP 90 = 2s，反之，只有当并发压力不高于 200 万时，我们才可以做出承诺。

具体应该如何考虑输入限制呢？有两大维度我们要去重点评估和考量，一类叫做业务限制，一类叫做技术限制。

业务限制主要包括以下六点：

第一，Time（时间）、Resource（资源）、Scope（范围）三要素。

这三类要素构成了产品最主要、最基本的“输入”，它们就像一个等边三角形，撑起了产品的落地路径。其中，时间和资源的含义都比较明确。范围一词略微有些模糊，我们将其理解为产品的功能规划或能力范围。范围出现变化，时间和资源也会相应出现变化。

最可怕的事情，莫过于像我们文章开头所举的例子一样，一个月的工作量要求一周做完、十个人的工作量要求用一个人做完、产品的范围被扩充成“航空母舰”。只要三要素不合理、不匹配，产品基本都会“难产”。

第二，法律法规与政策限制。

这两年的 P2P、金融创新、社区团购等业务可以算是典型案例。但法律法规和相关政策的变更，大部分是为了在国家角度控制市场风险。对于相关行业从业者来说，不应该视为障碍。就像我们先前讲过的，**好的产品设计是向善的，这是始终不变的。**

第三，组织文化限制。

不同的组织结构、组织文化，都会对产品的输入造成限制。这也是管理者为什么要聚焦「管理三板斧」，建立好的企业文化 —— 没有合适的组织，一切设计都是空谈。

第四，地域因素限制。

不同地域也会对产品落地造成限制，但可能不会造成直接影响。比如具有高技术壁垒的底层基础设施产品，如果放在三线城市进行孵化，就会面临团队人才不足的情况；特定项目在上海临港新片区孵化可能会相对容易，在其他城市可能就会相对困难。这些都是客观存在的。

第五，风险承受能力。

举例来说，薄利多销且过于依赖场景的业务，风险承受能力可能就会有所欠缺。比如今年，疫情对许多线下业务几乎造成了毁灭性的打击。

第六，市场因素限制。

市场因素也是我们要重点考虑的，这就需要对市场动态保持一定的敏感度，不能闭门造车。

除了业务限制，还要考虑**技术限制**，具体包括五类限制因素：

第一，遗留系统限制。

企业在系统建设的过程中，会引入一些商业套件软件或者自研系统。但同时，数字化资产管理可能没有做到位，导致很多设计文档缺失，相关负责人也不断变动。所以，后来人员很难再对系统进行升级。

究其原因，主要是架构设计缺乏、研发管理缺失。遗留系统越多，对于后来迭代开发速度影响越大。

新系统上线的时候，架构师的一个重要工作便是做系统的上下文架构设计、明确周边依赖，对于需要和周边遗留系统进行集成交互的组件或模块，要高度重视。这些都是关键的限制因素，可能因为一个小小的环节疏漏，就最终影响了工作的推进。

第二，团队技能限制

团队成员本身的能力，也是一大限制。一些团队刚刚经历了“大换血”，新成员需要成长。此时若要保证产品输出不变，势必需要在其他维度有所补充。

第三，现有基础设施限制。

前面我们讲过，企业的竞争壁垒来自于产品，IT 基础设施也是产品。基础设施强，团队能力就强，输入限制就少；反之，基础设施弱，团队能力就弱，输入限制就大。

比如资源扩容的时间、访问数据服务的复杂程度、测试回归的能力等，都是非常重要的限制因素，对于研发速度、研发质量都有巨大的影响。

第四，标准规范限制。

一般来说，随着企业 IT 治理水平的提升，都会逐步建立企业范围内的标准规范。这些规范是企业最佳实践的总结、规划，但同时也对产品建设构成了限制。

比如，一些企业会对研发语言有所限制。在某些情况下，精通 Java 语言的可用工程师比较少，但仍然要求用 Java 语言完成开发，这也会对输入产生限制。

第五，实施限制。

在实施方面，企业往往也会有所限制。比如，流程性的规定一般会对研发时间有所限制，做排期时需要纳入考虑。

你看，一个产品从设计到落地，会受到如此多的输入限制。如果不考虑它们，一定会对“成功交付”造成很大的影响，最终影响工作目标的达成。

讲完输入限制，我们再来理解一下所谓的输出限制。

输出，是给用户的契约，我们在产品思维、高可用设计等章节都有所提及 —— 要给出清晰、可量化的、符合 “SMART 原则”的契约。可以说是“对契约做限制”，也可以理解成“要让契约更明确”。

比如，在对系统进行容量规划后，架构师要对超出处理能力的流量进行流量控制。这就是一种明确的限制手段；再比如，如果机房只有一根光纤，就不要承诺系统 7 * 24 小时高可用；如果生鲜产品需要早上 7 点送达目标企业，那么原则上，晚上 10 点后就不再接单……

这些都是限制，需要将其加入服务契约中，和用户形成共识，一诺千金。明确限制，是为了更好地服务用户，这是对用户负责的态度。

产品迭代背后的项目管理

在实际工作中，每个技术人员都会通过做项目的方式，参与产品的迭代。而做项目本身，就是一场关于“限制”的重头戏。

你可能会想，做项目有什么可说的？我们的项目经理每天就会催、催、催，这活我也能干。

其实，项目管理是一门专业度极高的学问。很多项目看似是因为一些意外情况延期、取消，实际恰恰是项目管理做得不够专业。

在一个项目组里，有四类角色非常重要，分别是：

PM（Project Manager），项目经理；

PM（Product Manager），产品经理；

Architect，架构师；

Specialist，某一领域技术专家，比如 DBA ，分布式缓存专家，大数据专家等。

在大型项目里，这四类角色可能分别对应着四个人，甚至更多；在中小型项目中，四类角色也可能只对应着一个人，这意味着，团队内有一个成员很强，可以同时承担这四类角色的职责。

项目的成败，整体掌握于项目经理之手，因为项目经理要做好 WBS（工作分解结构）。项目在落地过程中，所有的节点监控、风险管理等工作，都依赖于 WBS 。

WBS 有三大分解原则：

1. 将主体目标逐步细化分解，最底层的日常活动可直接分派到个人去完成；
2. 每个任务原则上要求分解到不能再细分为止；
3. 日常活动要对应到人、时间和资金投入。

前面我们讲了，业务因素和技术因素都会对输入产生限制，但具体如何对时间、人力、资源产生影响，影响程度怎么样，可不是拍脑门决定的。

现在，很多项目经理做工作拆分的流程是：将所有相关人员叫到一起，分别询问各工作项的预估执行时间。可能各模块负责人自己也不太清楚，秉持着“ Timeline 要虚报”的原则，直接拍脑袋给了个上浮过的整数时间，比如：一个月、两个月……项目经理说，好！于是 Timeline 就这么确认了。

这就是典型的非专业项目落地，最终不但造成项目时间的不可控、项目执行的低效，长期来看，也导致了业务各方、企业各层间的博弈，比如，老板会说，你们能不能更快点？你们是不是在“放羊”？

有经验的项目经理在预估排期时，往往会问每个模块的负责人：“为什么这些工作需要 x 天的执行时间？有没有什么执行难点？”如果对方答不上来，或者该负责人也没有相关的项目经验，项目经理就需要找到有经验的人、有发言权的人，继续问：“这些工作需要多久做完，为什么？有没有什么执行难点？”

看起来并不复杂，但实际上，是需要专业能力支撑的。项目经理凭什么做拆解，并确定任务间的依赖关系？答案是，要么组织产品经理、架构师、技术专家，一起分解任务；要么一人身兼四种角色，独立完成。

WBS 制定完成后，产品经理以产品设计匹配业务需求，明确业务目标和业务流程；架构师负责整体的框架设计，明确对应组件的整体视图；技术专家负责扫清相关的技术难点。最后，Developer（开发人员）、Tester（测试人员） 进入项目，解决工作量问题。

到头来，我们要明白，以上工作都是在输入、输出受限的情况下，保证产品不会因为执行问题而违背契约。反过来讲，执行团队本身也是输入的限制条件。也就是说，当缺乏优秀的项目经理、产品经理、架构师或技术专家时，就要相应地继续下调输出预期。

而且，**项目同样受到 Time（时间）、Resource（资源）、Scope（范围）等多方因素限制，在执行时同样要考虑限制条件，其本质就是有“不完美”成分存在的。通过不断的迭代，产品最终变得完美，但项目不会一次就缔造一个完美的产品。对项目期望的合理管控，本身就是在考虑限制。**

在做项目执行时，认识到这一点是非常重要的。

延展思考：向上管理的“限制”问题

把控好了输入、输出的“限制”，也做好了项目执行过程中的“限制”，基本上就覆盖了产品从 0 到 1 的整个流程。

理论上讲，我们的产品应该不入险地，始终“高可靠”地完成契约交付。

但实际情况未必如此。

问题出在哪里呢？一个意外是，项目经理在做输出限制时，通常会遇到来自上层的阻力。

比如，项目经理要立项研发企业 OMS 系统，但企业基础设施建设程度一般、团队技术实力较差，因此输出的契约是“三个月上线 OMS 系统”。老板看到了，可能就会问：“怎么这么久，一个半月行不行？”

这时，一个很常见也很难解的命题就出现了：如何做好向上管理？

在前面的文章里，我们其实分享过，要做好向上管理，第一要务是具备全局思维能力，和老板站在同一个维度思考。所谓的沟通技巧，只能锦上添花，不能雪中送炭。

很多同学当时很有感触，但在沟通中，我发现，大家的认知可能还有偏差。全局思维，是为了对齐目标、统一话语体系，但在具体表达的时候，一定要体现自己的专业性。

你的结论是：OMS 系统三个月上线；老板的想法是：OMS 系统一个半月上线。此时，你第一件要做的事，就是利用自己的专业知识和行业一般数据，将与 WBS 有关的思考和推断表达出来，尝试让项目按照正确的节奏落地。

第二点同样重要：要从更快交付的目标出发，同时将多种限制因素考虑在内，比如团队疲劳度、凝聚力、激励措施等，和老板一起想办法，把自己想象成一个专业的 CEO —— 这公司就是自己的，进而思考应该如何安排工作、如何制定 WBS。

你可能会说，老乔，那如果讲不清楚，应该怎么办啊？其实没啥办法，如果你不够专业，或者讲不清楚，那也只能认了。所以我对中层管理者的能力考察，包括：专业能力、汇报能力 —— 技术人不但要能干活，还要能专业地讲出来。

最后，要千万注意：**具备全局思维、让自己变得更专业、提升自己的表达能力，这一切工作都不是为了和老板在项目目标上进行博弈。我们的目的是，通过更合理的实现路径，最终达成业务的增长目标。**否则，除非你非常优秀，是不可或缺的人才，不然一般都会在博弈中处于下风，对个人成长造成影响。

结语

今天，我们聊了聊面对产品，如何做好限制，尤其强调了对输入、输出以及项目执行的限制。在架构设计、高可用、高性能等内容里，我们无时无刻不在强调限制。

很多同学，尤其是比较有上进心的同学都会想，强调这个干嘛呢？无论是什么工作，尽力去做不就好了吗？做成什么样算什么样，反正尽力了。

其实，恰恰是因为有上进心，才应该尽全力交付为用户承诺的契约，所有组织最终都是结果导向的。产品有限制，说明当下仍有许多不足，知不足，才能知进步。

限制，在企业层级，也意味着在企业发展的某个阶段，选择放弃某类用户。有放弃，才能聚焦，进而倒逼自己看到全局，和企业目标对齐。聚焦当下，学会取舍；放眼长期，持续进步、持续改进。

到这里，关于如何做好限制，就基本讲完了。我们的专栏，也正式进入了完结倒计时。还是老样子，如果有问题，欢迎在评论区提问。

我们下一讲再见！

提建议

训练营推荐

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 22 | 扩展性设计，看透业务的本质

下一篇 24 | 监控设计，让一切都有迹可循，尽在掌控

精选留言 (8)

写留言

spark
2020-12-16

我一般能用计划的30%时间完成我负责的任务，其他时间要么没事干、要么项目的代码都是我在Review，其他兄弟们加班累死累活、延迟交付且质量不高。
我做事的方法：
1.向领导学习(让领导替我想想办法如何完成的快和好)：做事前对齐目标，路径。和领导沟

通如何更快完成任务，领导的效能至少比我高3倍5倍。这样就算最后完不成，领导也不...
展开

作者回复: 你好，spark
你太棒了，真的很棒，领导应该提拔你才对。

1 3

Weehua
2020-12-18

我作为技术管理者，感受最大的就是项目排期，总是和业务方或者老板博弈。如果坚持自己原定的排期，有些重要需求确实对业务来说很重要（比如双11或双12搞活动），这个时候只能压缩排期，只能加班了，但确实有些需求不符合客观规律，或者如果按时交付但质量不行也经常出现，最后会损害团队声誉，有时候也是各方逼迫的。
...
展开

作者回复: 你好，Weehua
从业务发展出发，和业务一起讨论共赢方案。考虑限制，和业务讨论限制，是对业务更负责任的态度，相信时间会证明自己，业务会理解，双方的信任也会越来越强。

1

流浪地球
2020-12-17

请教乔老师一个问题，文中提到了项目经理在一个项目成败中的重要性，自己感同身受，有个问题是项目经理在一个团队中的职权范围应该是怎样的，或者更直接的说地位应该摆在什么高度。经历过的2个团队
1、pm职权范围极大，在制作人一人之下，对整个团队成员有考核权。但问题是这个角色本身的技术专业度不够深，和技术leader，产品leader很容易产生摩擦，团队成员往往...
展开

作者回复: 你好，流浪地球
实际是靠团队。
在项目启动中，一个项目要明确PM-项目经理，PM-产品经理，架构师，专才，项目启动就是明确各自职责。
从公司层级，项目启动的时候，项目所有人要听PM的安排，PM要充分发挥以上四种角色的专业，一起制定项目的WBS。
所以权力在项目启动的时候赋予了项目经理，项目启动会其中一个最重要的目的是赋予项目经理权力，否则那不是项目经理，而是项目助理。

1

glutton
2020-12-16

乔老师讲的，总能让我有通透的感觉
放在个人成长上，"知不可为"(考虑限制)可能比"知可为"更重要。之前一直忽略了这点。
又值得反复读的一课，感谢乔老师持续输出高质量干货。

展开

作者回复: 你好，glutton
谢谢，感谢你的陪伴

1

术子米德
2020-12-20

🤧☕🤧☕🤧
有一种限制，也很致命，那就是大家对于项目本身可能遇到的问题的认识，这个认识本身就是一个关键约束
如果遇不到啥大问题，大概率这个项目本身比较平庸；如果这个项目有挑战，那么很多问题就是在项目开展过程里，才会遇到、才开始真正处理到挑战点。这时候往往出现认知...

展开

作者回复: 你好，术子米德
你提的这个问题的解决，要在管理三板斧的目标对齐解决，如果目标对不起，专业复盘中提的问题都没有根。
专业会变成出现问题时扯皮的资本，根基不稳，一切都没了基础，这也是专栏组织的一个考虑点。

garlic
2020-12-17

项目外包，为甲方做的产品，我觉得外包工作还有一个限制就是要考虑公司收益最大化，我入行的时候经常听到的就是“给多少钱，办多少事”，如果换个角度，就是把甲方的产品做好，有源源不断的需求，从而对公司产生更大收益。再换个角度，我是甲方，产品还有不错的发展，那我就直接自己干了。

展开

作者回复: 你好，garlic
说清楚限制，更是一种负责任的态度。

子夜枯灯
2020-12-16

老乔讲的内容干活满满。化为己用，会是听完课程优先需要考虑怎么落地首要问题。

作者回复: 你好，子夜枯灯
谢谢，学以致用很重要，都掏钱学习了，一定要用起来

土圭垚
2020-12-16

我一定要坚持看完，毕竟钱花了不给退
展开

作者回复: 你好，土圭垚
哈哈，坚持坚持，是的，都花钱了，一定要看完，并且应用起来，最大化学习的价值。
持续学习，学以致用，加油